

Chapter 2: Communication in distributed systems.

* What is the most important difference between distributed system and uniprocessor system?

↳ It is the inter~~process~~^{process} communication.

a) in uniprocessor, most interProcess communication assumes existence of shared memory.

b) in distributed, no shared memory so entire nature of interProcess communication must be completely rethought from scratch.

InterProcess communication

↳ N Processors each with its own data stream.

↳ used to communicate data between Processors.

This is done by

1) using shared memory & shared variables.

2) via an interconnection network.

I] Using shared memory & shared variables

→ This consists of global address space which is accessible by all N Processors

→ Processor can communicate ~~by~~^{to} another by writing into global ~~address~~ memory where 2nd Processor can read it.

*role of shared memory

↳ solves the interprocessor communication problem but introduces problem of simultaneous accessing of the same location in the memory.

ملف 3 من 4 (slide) للقاء ٥

*shared memory

→ each processor has access to any variable residing in shared memory.

if processor x wishes to pass a number to processor y this happens in two steps:-

- 1) x writes the number in shared memory at a given location accessible to y.
- 2) processor y reads the number from that location.

Notes

a) During execution of parallel algorithm, N processors access shared memory for reading/writing data and/or results.

b) All processors can gain access to shared memory simultaneously if memory locations they are trying to read or write are different.

c) When two or more processors require access to same memory location simultaneously, we can get problems.

* subclasses of shared memory computers

1) Exclusive Read, Exclusive write (EREW) ~~computers~~

→ Access to memory locations is exclusive so no 2 Processors are allowed to simultaneously read from or write into same location.

2) Concurrent Read, Exclusive write (CREW)

→ Multiple Processors allowed to read from the same location but write is still exclusive.

→ ~~no~~ 2 Processors allowed to write into same location simultaneously.

3) Exclusive Read, Concurrent write (ERCW)

→ Multiple Processors are allowed to write into the same memory location.

4) Concurrent Read, Concurrent write (CRCW)

→ Both multiple read and multiple write are allowed.

⇒ Each of several Processors reading from that location makes a copy of its contents and stores it in its own register (RAU)

→ Problems arise however, with Concurrent write access:

↳ if several processors are trying to simultaneously store data at the same address, which of them should ~~success~~ succeed?

⇒ We need a deterministic way of specifying the contents of a memory location after a concurrent write operation.

* Some ways of resolving write conflicts include:

- 1) Assign Priorities to Processors and accept value from highest Priority Processor.
- 2) All Processors are allowed to write, the quantities they are attempting to store are equal, access is denied to all Processors.
- 3) The max / min / sum / average of value is stored.

* why shared memory SIMD machines are ~~unrealistic~~ unrealistic?

↳ ~~Because~~ ~~PP~~ because the cost of communication Hw is too high and physical size of the device used for a memory location.

⇒ In MIMD (which use more powerful processors) shared memory is existed which have small number of processors (2-30)

هنا يوجد مثال لتوضيح الأربع أنواع السابقة في صفحات 10, 11, 12
- slide 2 3

InterConnection network and message Passing

- each Processor has its own private memory and there is no global, shared memory.
- Processors need to be connected in some way to allow communication of data to take place.

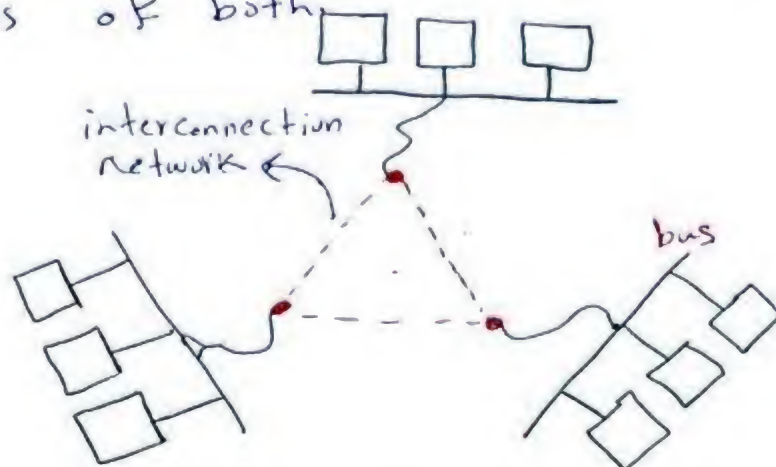
Example if processor require data contained in separate processor then it must be explicitly passed by using instructions supplied for communication e.g. send and receive functions.

• P_1 P_2
send(x, P_1) receive(x, P_2)

→ value of x passed from P_2 to P_1 .

↳ this is known as message passing.

- there is possibilities for hybrid designs that combine features of both



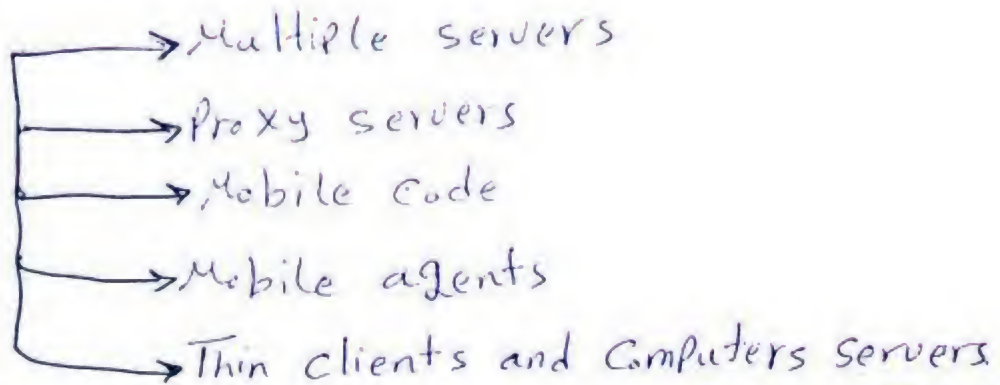
← مثال توضيح الفرق بين shared memory, distributed sys.

من صفحة 16 لصفحة 19 (3 Slide 2.

* Architectural models of multiComputers

1) client-server-model 2) Peer-to-Peer model.

3) Variations



* What is the difference between these models?

- 1) Placement of components (data & workload distribution) across the network Computers.
- 2) The interrelationship between components (role, of each one & pattern of communication).

1 client-server-model (Most widely used)

a) server

- service SW running on single machine.
- Process in networked Computers that accepts requests from other processes to perform a service and responds in a proper way.

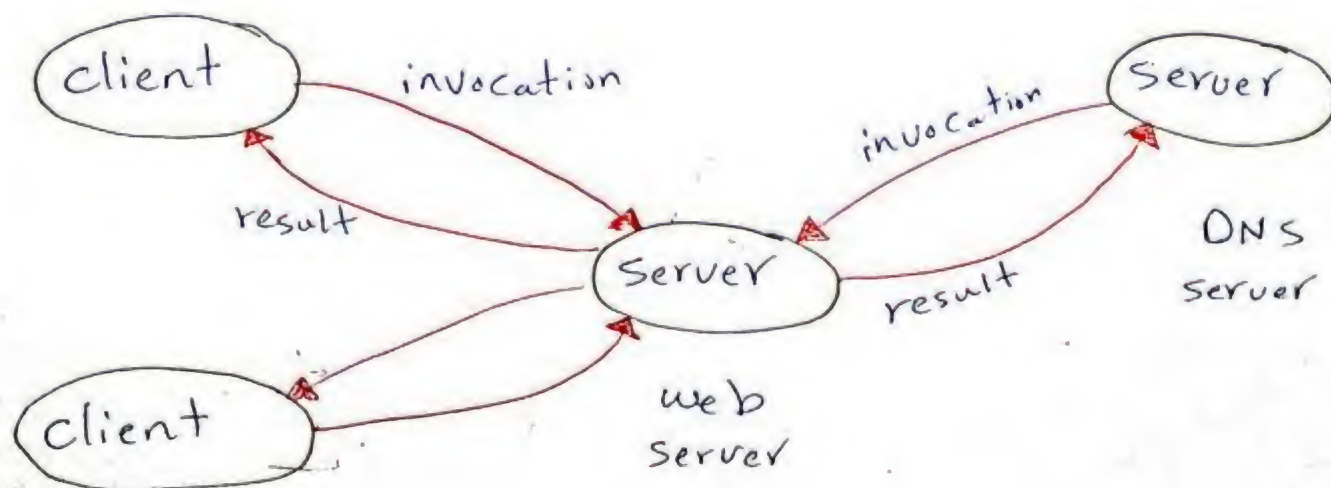
b) client

- Process that can invoke a service using a set of operations that forms its client interface.
- requesting process in the above is referred to as the client.

Notes

- Request & response are in form of messages.
- client is said to invoke operation on server.

- * Can servers be clients for other servers?
- It may be happen e.g. a web server is a client of a local file server that manages the files in which the web pages are stored
- web servers and most other internet services are clients for DNS service which translates the internet domain names to network address.



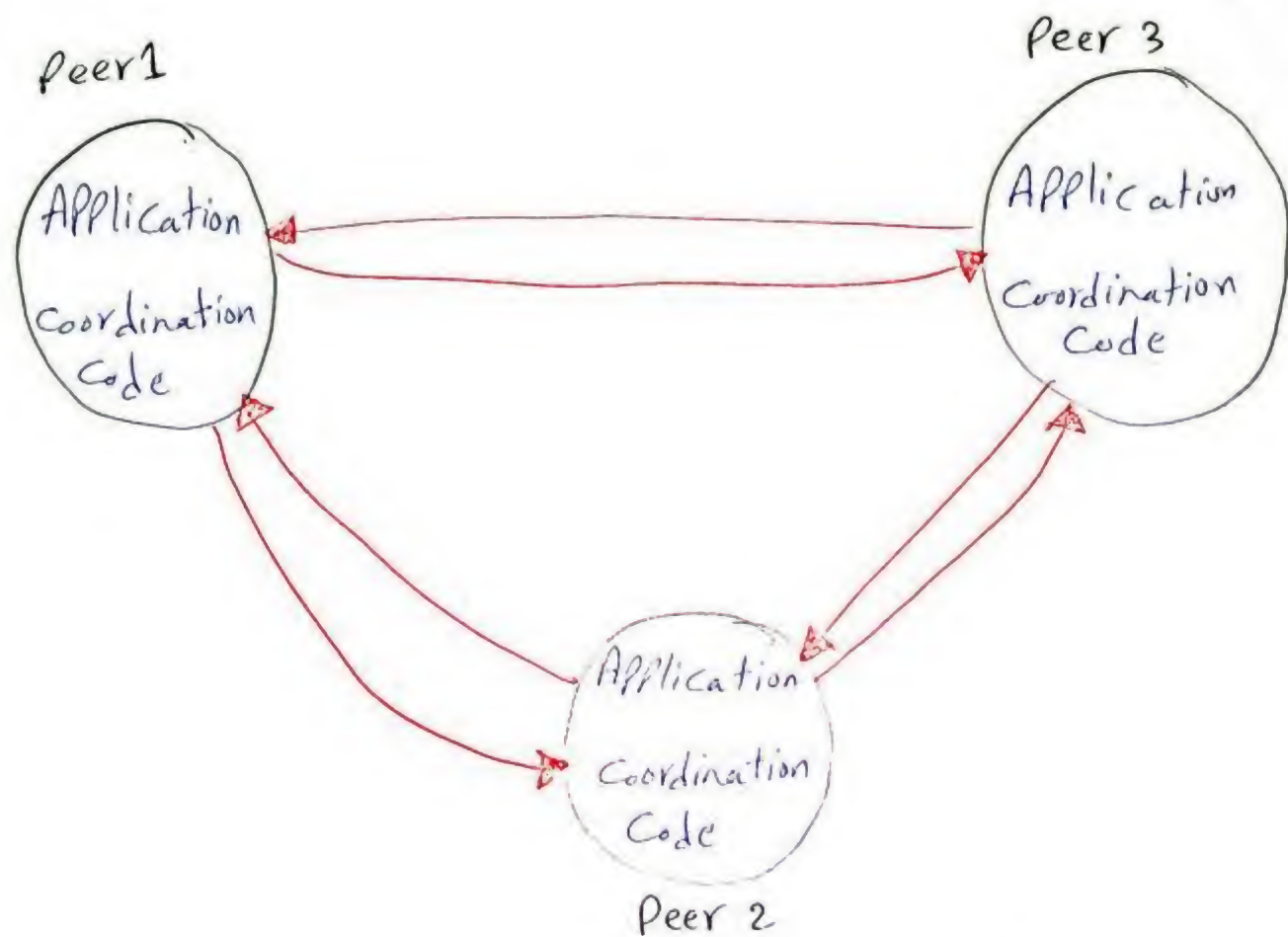
* Domain name system

- distributed ^{database} ~~system~~ implemented in a hierarchy of many name servers.
 - application layer protocol that is responsible for resolving names (address/name translation)
-

2) Peer-to-Peer model

- All processors play similar roles, interacting as peers (with difference between server and client processes or the computer they run on) to perform tasks.
- * What is the problem with client/server?
 - It is a scale up problem, centralizing the services at the server makes it overload under large number of clients.
 - the overload value is determined by the server's capability and network bandwidth.

So, there is a need to distribute shared resources ~~in~~ to share computing and communication loads to large no. of computers and network links.



→ what is the aim of P2P arch ?

↳ to make use of the resources (HW or data) of participating computers to fulfill a given task or activity.

What is Napster

→ the most famous Peer-to-Peer instance.

→ application that facilitates that sharing of music and video recording among its users around the world.

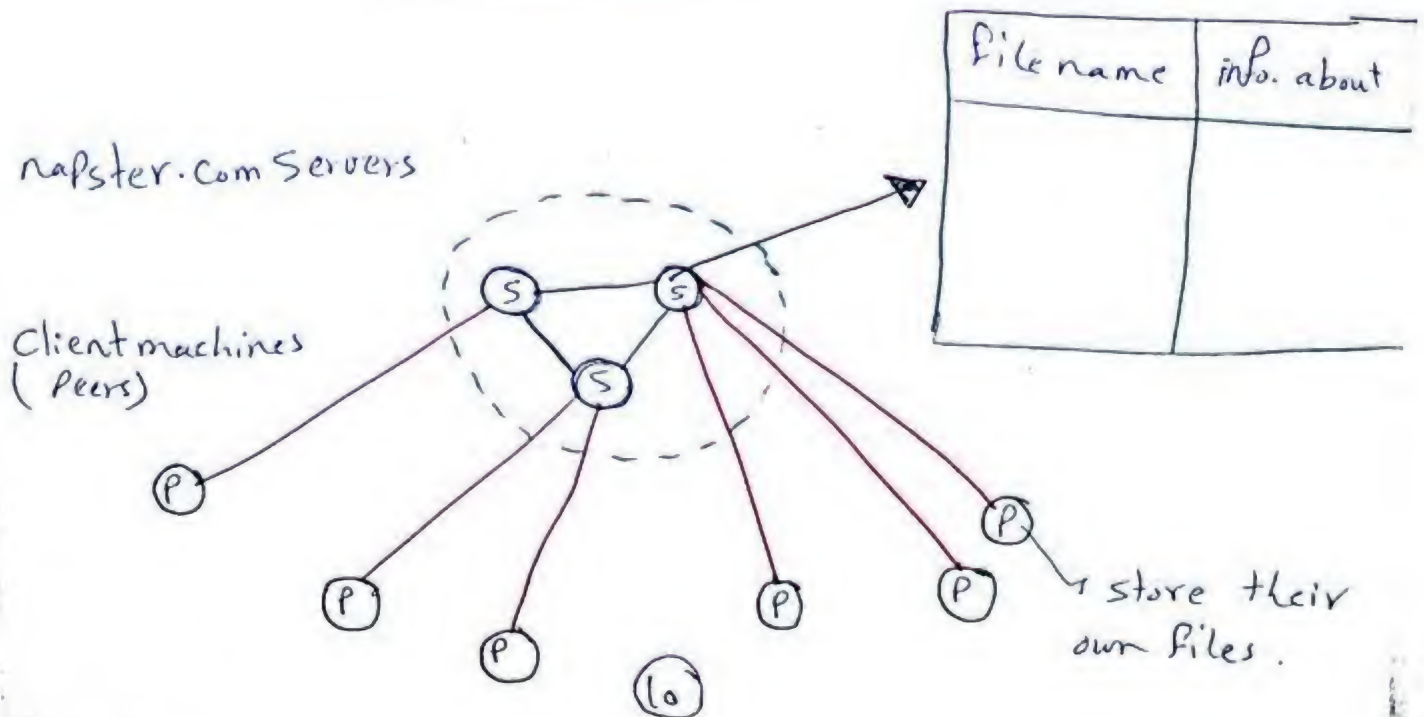
* client

- Connect to a Napster Server.
- upload list of music Files that you want to share (names only, not the Files)
 - ↳ server maintains list of
 < Filename, ip-address, Portnum > tuples.

* search From a client

- Send Server Keywords to search with.
- Server Searches its directory with Keywords.
- server returns a list of matching hosts.
- client pings each host in list to find transfer rates
- " fetches File from best host.

Napster structure



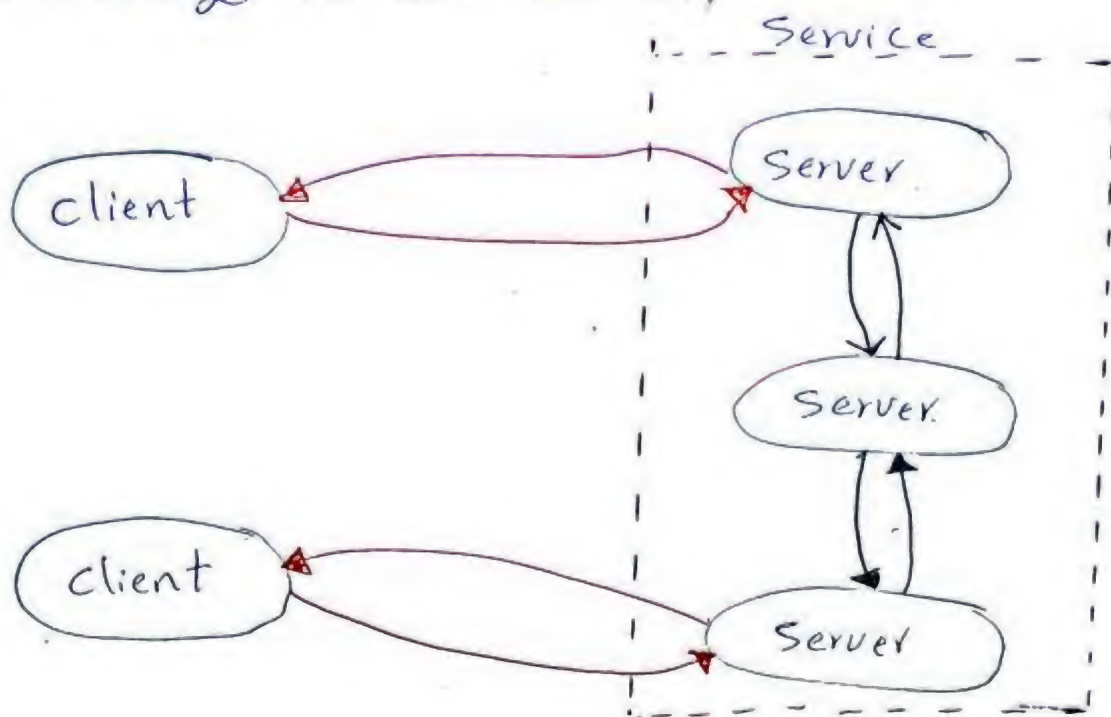
3.a) Multiple Servers

→ Services may be implemented as several server processes in separate host computers interacting to provide service to client processes.

→ Server may partition the set of objects on which services is based and distributed on several hosts

↳ this can be illustrated by example:

⇒ web is good example of partitioned data which is distributed on different hosts with each server can manage its own set of data files.



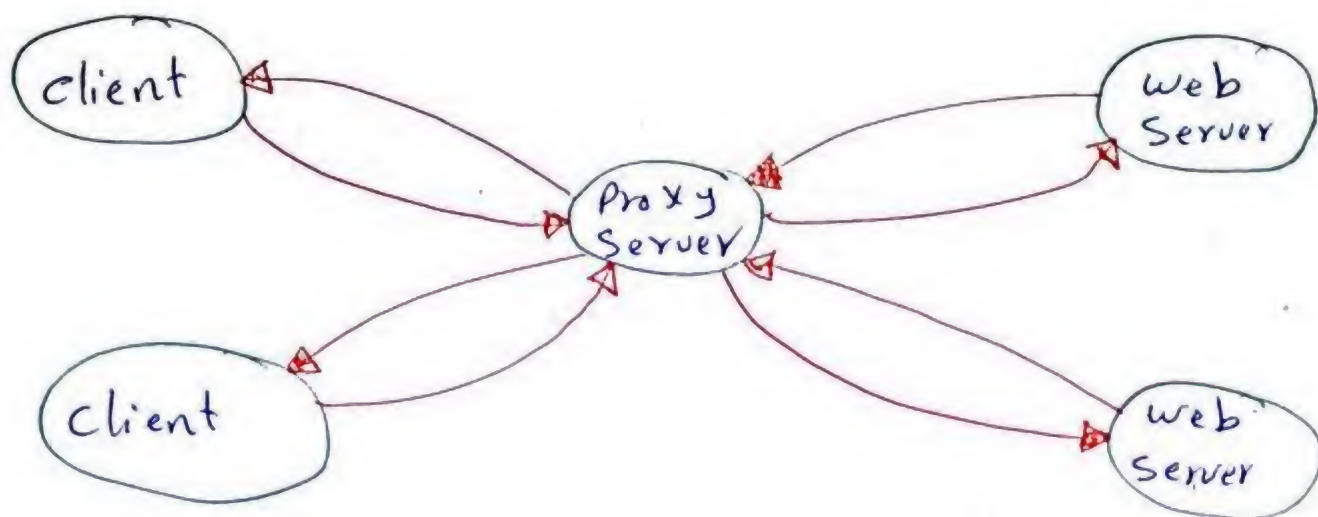
- Cluster is example of multiple-server architecture which is used for highly scalable web services.
 - Service Processing can be partitioned or replicated between them.
-

3.b) Proxy server

- It provides a shared cache of web resources for client machines at site or across sites.
- Cache: store of recently used data objects.
- * Different situations for object?
 - 1- When new object is ~~needed~~ received at a PC it is added to cache (if cache is full, object will replace other objects)
 - 2- When object is needed by client process, Caching service checks the cache and supplies it if it is available and up-to-date.

Note

- ↳ Caches is located at proxy server that can be shared by several clients.



* what the Purpose of Proxy server?

- a) increase availability and Performance of services by reducing the load ^{on} WAN and web servers.
 - b) may be used to access remote web servers through Firewall.
-

3.c) Mobile Code

→ example For it applet

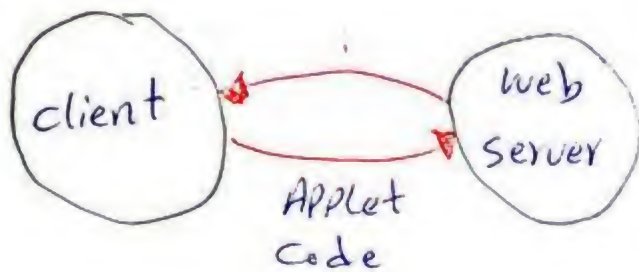
↳ user is running a browser selects a link to an applet whose Code is stored on web server.
↳ then Code is downloaded to user's browser and runs on user's side.

⇒ It may be called Code migration, why?

Cause moving From heavily loaded to lightly loaded system (server and network)

its steps

a) client request results in downloading of applet Code



b) client interact with the applet



* Advantage of mobile Code?

↳ it gives good interactive response ~~time~~

* it doesn't suffer from delays

* " " " " variability of BW due to variable network loads.

* notify us with new versions of services such as virus scan Programs.

* what is the meaning of Push model?

↳ It is the interaction between client/server as the client should have initiated the communication with the server each time.

* Is the applet dangerous to local resources of client's computer? if ^{why &} yes? what is the solution?

→ ~~stockbroker might provide~~

→ client has to download the special applet that receives the updates from the broker's server and display them for client.

↳ this rises security problem, so yes it is very dangerous -

Solution browsers give applets limited access to local resources (like: not allowing it to access local files)

3.d) Mobile agent

→ is running program (include code and data) that travels from computer to another in network carrying out task and return with results.
example of tasks: collect information or search.

→ It converts the remote invocation into local invocation which means reduction in communication cost and time.

* uses of mobile agent

- a) install and maintain SW on Computers in organization
- b) Compare Prices of products from no. of vendors by visiting the site of each vendor and perform some DB operations.

What is the worm Program?

→ older version of mobile agent.

→ It was designed to make use of idle Computers to carry out intensive Computations.

* mobile agent rises security problem So, visited Computers that receive mobile agent should decide which local resource should access according to ~~it~~ identity of user initiating ~~to~~ this agent.

⇒ What will happen if mobile agent is denied of access from resources it needs?

↳ then it may fail to complete its work.

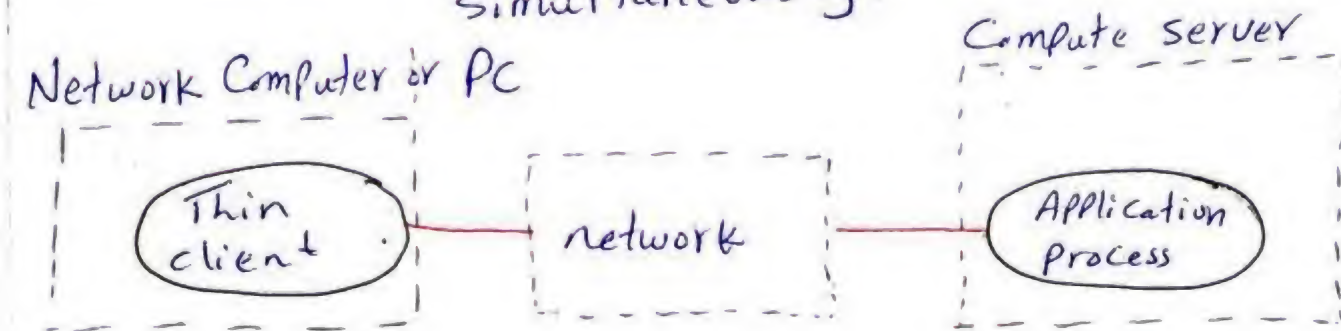
So applicability of mobile agent is limited.

3.e) Thin client / Compute server

→ Thin client is SW layer that supports windows-based user interface on computer who is local to user while executing application programs on remote computers.

→ code of apps. is runned on Computer server.

* Compute server is Powerful Computer that has the capacity to run large numbers of apps. simultaneously.



* What is the ~~main~~ main advantage of this model?

↳ when it is used in highly interactive graphical activities (such as image processing) where user suffer from delay, ~~due to~~

Types of distributed operating systems

[1] Network operating systems:- (such as Unix & windows)

- they have built-in network capabilities
- they can use this network to access remote resources.
- users are aware of multiplicity of machines.

File unity of communication between machines.

* How can access of resources of various machines is done?

- remote logging in appropriate remote machine.
- remote desktop (Microsoft windows)
- transfer data from remote machines to local machines.

[2] Distributed operating systems (Dos)

- users are not aware of multiplicity of machines.

* Different tasks provided by Dos

1) Data migration

- ↳ transfer data by transferring entire file.
- or by transferring partitions of file needed for immediate task.

2. Computation migration

↳ transfer Computation rather than data ^{across} ~~across~~ System.

3. Process migration

↳ execute an entire Process or Parts of it at different sites to achieve: ~~these following~~:

- a. **load balancing**: distribute Process across network.
- b. **Computation Speedup**: subprocess can run concurrently on different sites.
- c. **HW Preference**: Process execution may require specialized processor.
- d. **SW Preference**: required SW may be available at only a particular site.
- e. **Data access**: run Process remotely

Communication structure

address four issues:

1. Naming and name resolution (~~How do the process~~)
2. Routing strategies.
3. Connection strategies.
4. Contention.

[1] Naming and name resolution

↳ How two processes communicate in network?

⇒ We have many ways:

1) Address messages with Process-id.

2) identify processes on remote systems by
 <host-name, identifier> pair

3) Domain name service (DNS) ⇒ specifies naming structure of hosts, like name to address resolution.

[2] Routing strategies

→ How messages sent through network?

a. Fixed routing

→ Path from A to B is specified in advance.

→ Path changes only if HW failure disables it.

→ Communication costs are minimized (shortest path)

→ It cannot adapt to load changes.

→ Ensure that messages will be delivered.
 in the order in which they were sent.

*Virtual circuit

↳ Path From A to B is fixed for the duration of one session.

⇒ Different sessions may have different paths.

a. Partial solution to adapting to load changes.

b. ensure that messages will be delivered, in the order which they were sent.

[b] Dynamic routing

↳ Path used to send message from site A to site B is chosen only when message is sent.

→ usually site sends message to another site on link least used at particular time.

→ Adapts to load changes by avoiding routing messages on heavily used path.

→ its Problem message may arrive out of order?

↳ it can be solved by appending a sequence number to each message.

3) Connection strategies

↳ How do two processes send a sequence of messages?

a. Circuit switching : Permanent Physical link is established for duration of communication.
(telephone system)

b. Message switching (Post-office mailing system)

↳ temporary link is established for duration of one message transfer.

c. Packet switching message of variable length are divided into packets

→ each packet may take different path through network

Note

→ circuit switching requires setup time, but incurs less overhead for shipping each message.

→ packet & message switching require less setup time, but incur more overhead.

[4] Contention

→ network is shared resource, so how do we resolve conflicting demands for its use?

→ several sites may want to transmit information over a link together.

*Techniques to avoid repeated collisions

1) CSMA/CD

CSMA → Carrier Sense with multiple access

CD → Collision detection.

→ is used successfully in Ethernet system, the most common network system.

a. site determines whether another message is being transmitted over link.

b. if two or more sites begin transmitting at the same time, then they will register CD and stop transmitting.

c. When system is very busy, many collisions may occur and performance may be degraded.

2) Token Passing

→ unique message type, known as a token.

a. site that wants to transmit information must wait until the token arrives.

b. When the site completes its round of message passing, it ~~retransmits~~ retransmits the token

c. A-token-passing scheme is used by some IBM and HB/Apollo systems.

[3] Message slots

→ number of Fixed-length message slots continuously circulate in the system.

a. since slot can contain only Fixed-size messages, single logic message may have to be broken into smaller or smaller packets, each is sent in separate slot.

b. This scheme has been adopted in the experimental Cambridge digital communication ring.

[24]